Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет центр післядипломної освіти
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів впровадження та підтримки інформаційної безпеки при використанні DevSecOps практик в організації розробки IT-стартапів
(тема)

Виконав:
студент 2 курсу, групи ІПЗздм-22-1

Приходько Я.О.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник доц.кафедри ПІ Лановий О.Ф.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ З.В.Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет центр післядипломної освіти
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Приходько Яну Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів впровадження та підтримки інформаційної безпеки при використанні DevSecOps практик в організації розробки IT-стартапів»
Затверджена наказом по університету від 22.04.2024 р. № 60Стз
2. Термін подання студентом роботи до екзаменаційної комісії 07.06.2024
3.Вихідні дані до роботи науково-методична та науково-технічна література, дані з інтернет джерел, статті, щодо проблем інформаційної безпеки стартапів та DevSecOps практик та інстументів, база вразливостей CVE
4. Перелік питань, що потрібно опрацювати в роботі
Аналіз впровадження DevSecOps практик, реалізація моделі ШІ за допомогою Python, датасет CVE, фрагменти коду, графічні матеріали.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	23.01 – 14.02.24	*виконано*
2	Аналіз ключових викликів інформаційної безпеки для IT-стартапів	15.02 – 24.02.24	*виконано*
3	Дослідження імплементації інформаційної безпеки з використанням методів DevSecOps	17.02 – 28.02.24	*виконано*
4	Планування та підготовка даних для моделі дослідження	25.02 – 28.02.24	*виконано*
5	Побудова моделі для методів та інструментів DevSecOps	25.02 – 01.04.24	*виконано*
6	Аналіз результатів дослідження та оцінка актуальності методів DevSecOps	02.04 – 20.04.24	*виконано*
7	Підготовка рекомендацій щодо застосування та висновків за аналізом	20.04 – 23.04.24	*виконано*
8	Написання статті та тез доповіді	17.04 – 23.04.24	*виконано*
9	Підготовка пояснювальної записки	01.04 – 26.04.24	*виконано*
10	Підготовка презентації та доповіді	26.04 – 5.05.24	*виконано*
11	Нормоконтроль	5.05 – 08.05.24	*виконано*
12	Рецензування	08.05 – 11.05.24	*виконано*
13	Занесення диплома в електронний архів	12.05.2024	*виконано*
14	Попередній захист	13.05.2024	*виконано*
15	Допуск до захисту у зав. кафедри	15.05.2024	*виконано*

Дата видачі завдання 20 січня 2024р.

Студент (ка) _____ Приходько Я.О.
(підпис)

Керівник роботи _____ доц.кафедри ПІ Лановий О.Ф.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка містить: 80 с., 7 рис., 5 табл., 21 джерело.

АЛГОРИТМИ КЛАСИФІКАЦІЇ, ІНФОРМАЦІЙНА БЕЗПЕКА, COMMON VULNERABILITIES AND EXPOSURES, DEVSECOPS.

Об’єктом дослідження є процес забезпечення інформаційної безпеки в організації розробки IT-стартапів, включаючи виклики та загрози, які виникають внаслідок швидкого розвитку і обмежених ресурсів.

Предметом дослідження дослідження є методи та інструменти DevSecOps і їхній вплив на сучасні вразливості та забезпечення інформаційної безпеки в організації розробки IT-стартапів.

Метою роботи є дослідження методів та інструментів DevSecOps і їхнього зв'язку з конкретними сценаріями ризику для ефективного впровадження та підтримки інформаційної безпеки в організації розробки IT-стартапів.

Методи розробки та проектування включають аналіз проблемної області, вивчення джерел для дослідження DevSecOps практик, побудову та аналіз моделі класифікації вразливостей на основі зв'язку з інструментами та методами DevSecOps.

У результаті кваліфікаційної роботи розроблено код для класифікаційної моделі за методом Gradient Boosting Machines (GBM).

CLASSIFICATION ALGORITHMS, INFORMATION SECURITY, COMMON VULNERABILITIES AND EXPOSURES, DEVSECOPS.

The object of the study is the process of ensuring information security in the development of IT startups, including the challenges and threats that arise from rapid development and limited resources.

The subject of the study is the methods and tools of DevSecOps and their impact on modern vulnerabilities and the provision of information security in the development of IT startups.

The objective of this work is to investigate DevSecOps methods and tools and their connection with specific risk scenarios for the effective implementation and maintenance of information security of the development process of IT startups.

The development and design methods include an analysis of the problem domain, a review of sources to research DevSecOps practices, and the construction and analysis of a model for classifying vulnerabilities based on their correlation with DevSecOps tools and methods.

As a result of the qualification work, the code of a classification model based on the Gradient Boosting Machines (GBM) method was developed.

Заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу ElArKhNURE.

Я, Приходько Ян Олегович, студент гр. ІПЗздм-22-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів впровадження та підтримки інформаційної безпеки при використанні DevSecOps практик в організації розробки IT-стартапів», що буде представлена на екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

У сучасному інформаційному середовищі, де IT-стартапи стають ключовим елементом інноваційного розвитку, питання інформаційної безпеки набувають особливої актуальності та значення. Інформаційна безпека не лише захищає конфіденційні дані та інтелектуальну власність, але й впливає на довіру клієнтів, інвесторів та партнерів.

Актуальність даного дослідження обумовлена зростанням кількості IT-стартапів та їх важливою роллю у сучасному технологічному середовищі. Забезпечення інформаційної безпеки стає критичним чинником для їх успішного розвитку та конкурентоспроможності. Для впровадження та забезпечення відповідного рівня інформаційної безпеки в IT-стартапах, розробники все частіше звертаються до DevSecOps практик.

Метою цього проекту є дослідження методів та інструментів DevSecOps і їхнього зв'язку з конкретними сценаріями ризику для ефективного впровадження та підтримки інформаційної безпеки в організації розробки IT-стартапів.

У багатьох стартапах, особливо на початкових етапах розвитку, існує тенденція нехтувати питаннями інформаційної безпеки. Це може бути обумовлено швидкістю розробки, обмеженістю ресурсів або недостатньою обізнаністю у цій галузі. Багато стартапів часто відкладають питання безпеки на потім, що може призвести до серйозних наслідків.

Крім того, на початкових етапах розвитку більшість IT-стартапів не мають фінансових можливостей для найму внутрішніх експертів з інформаційної безпеки. Це створює додатковий виклик у забезпеченні адекватного рівня захисту даних та програм.

Безпека середовища, в якому відбувається розробка програмного продукту, безпосередньо впливає на якість та безпеку кінцевого продукту. Недостатньо захищене розробницьке середовище може створити вразливості та витоки даних у готовому програмному продукті, що, в свою чергу, може призвести до негативних наслідків для користувачів та репутації стартапу. Таким чином, забезпечення високого рівня інформаційної безпеки на етапі розробки стає критичним

чинником успіху та стійкості IT-стартапу на ринку.

У зв'язку з цим, важливо розглянути питання інформаційної безпеки у контексті IT-стартапів та визначити ключові виклики та методи, що допоможуть забезпечити успішний розвиток та збереження безпеки даних у цих організаціях. Враховуючи обмеження та виклики, якими стикаються IT-стартапи, практики DevSecOps можуть відіграти вирішальну роль у створенні безпечного програмного середовища від самого початку. Інтеграція заходів безпеки на ранніх етапах розробки дозволяє не лише ідентифікувати та усувати вразливості до того, як вони стануть серйозними загрозами, але й сформувати фундаментальне розуміння важливості безпеки командами розробників. Це створює основу для розвитку культури безпеки, яка є ключовим елементом успішної інформаційної стратегії стартапу. Використання автоматизованих інструментів і методів DevSecOps для тестування безпеки, сканування вразливостей, моніторингу змін у коді можуть значно знизити ризики втрати даних.

З метою підтримки інформаційної безпеки, IT-стартапи можуть впроваджувати різноманітні DevSecOps інструменти і методи, які не тільки допоможуть захистити середовище та процеси розробки, але й підсилять довіру з боку клієнтів та інвесторів. Створення безпечного продукту несе не лише технічні переваги, а й сприяє збільшенню комерційного успіху.

Об’єктом дослідження є процес забезпечення інформаційної безпеки в організації розробки IT-стартапів, включаючи виклики та загрози, які виникають внаслідок швидкого розвитку і обмежених ресурсів.

Предметом дослідження дослідження є методи та інструменти DevSecOps і їхній вплив на сучасні вразливості та забезпечення інформаційної безпеки в організації розробки IT-стартапів.

Методи розробки та проектування включають аналіз проблемної області, вивчення джерел для дослідження DevSecOps практик, побудову та аналіз моделі класифікації вразливостей на основі зв'язку з інструментами та методами DevSecOps.

Задачі дослідження включають:

– аналіз ключових викликів інформаційної безпеки для IT-стартапів;

– аналіз сучасних джерел, що містять відомості про загрози та вразливості;

– проведення теоретичного дослідження існуючих методів DevSecOps, аналізуючи публікації, статті компаній та спеціалістів з предметної області;

– розгляд прикладів успішного впровадження методів DevSecOps;

– вивчення інструментів для впровадження методів DevSecOps;

– проведення практичного дослідження загроз інформаційної безпеки на основі моделі даних CVE та інструментів DevSecOps;

– аналіз застосування моделі для реального датасету для визначення актуальності методів DevSecOps;

– розробку рекомендацій щодо застосування методів DevSecOps у IT-стартапах на основі практичного та теоретичного дослідження.

Це дослідження пропонує новий підхід до інтеграції DevSecOps методів у процес розробки програмного забезпечення для IT-стартапів, що враховує специфічні виклики та обмеження цих організацій. Вперше досліджено зв'язок між DevSecOps методами та конкретними сценаріями ризику для IT-стартапів на основі бази загальновідомих вразливостей.

Практичне значення роботи полягає у розробці рекомендацій щодо впровадження DevSecOps методів в IT-стартапах, що дозволить підвищити рівень інформаційної безпеки та конкурентоспроможності таких компаній.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз ключових викликів інформаційної безпеки для IT-стартапів

Аналіз ключових викликів інформаційної безпеки для IT-стартапів є надзвичайно важливою складовою успішного розвитку таких компаній. Це особливо актуально в сучасному інформаційному ландшафті, де IT-стартапи стають основними джерелами інновацій та технологічного розвитку. Все частіше стартапи стають об'єктом інтересу як з боку інвесторів, так і з боку кіберзлочинців, що ставить питання інформаційної безпеки на передній план. Важливість аналізу цих викликів полягає в тому, що інформаційна безпека не лише захищає конфіденційні дані та інтелектуальну власність стартапу, але й впливає на довіру клієнтів, інвесторів та партнерів. Безпека стає важливою частиною іміджу та репутації компанії, і її порушення може призвести до серйозних фінансових втрат і втрати довіри. Розглянемо основні виклики та їхню важливість у деталях:

– обмеженість ресурсів;

– брак експертизи;

– швидкість розробки;

– забезпечення довіри клієнтів та інвесторів;

– потреба у відповідності правилам і регулюванням.

Обмеженість ресурсів. Одним із основних викликів є обмеженість ресурсів. Через обмежені фінансові можливості та обмежений персонал, стартапи часто стикаються з труднощами у забезпеченні належного рівня інформаційної безпеки. За даними «TechCrunch», у багатьох стартапів це стає причиною неявки до питань безпеки, що може призвести до серйозних наслідків [1]. Багато IT-стартапів, особливо на початковому етапі, часто мають обмежені фінансові ресурси та обмежений персонал. Забезпечення належного рівня інформаційної безпеки може бути витратним та вимагати спеціалізованих кадрів. Важливо розглянути стратегії оптимізації та ефективного використання ресурсів для забезпечення безпеки.

Брак експертизи. Брак експертизи в галузі інформаційної безпеки є одним із

основних викликів, з якими стикаються ІТ-стартапи. Зазвичай, на початковому етапі розвитку стартапу, коли обмежені ресурси та фінансові можливості, компанії часто не мають в своєму складі спеціалізованих фахівців з інформаційної безпеки. Такі професіонали, як програмісти, розробники, дизайнери, маркетологи як правило є основними на початку існування стартапу і без них важко уявити розвиток інноваційного продукту. Проте, спеціалістів з інформаційної безпеки в їхньому складі часто бракує. Через відсутність цих спеціалізованих фахівців, стартапи можуть недооцінювати важливість безпеки та не бути готовими до потенційних загроз [2]. Їм може бути важко визначити ризики та вирішити проблеми, пов'язані з інформаційною безпекою, оскільки це вимагає специфічних знань та навичок. Це може призвести до недостатньої захищеності даних та вразливостей системи. З метою вирішення цього виклику, ІТ-стартапи можуть розглядати можливість залучення спеціалізованих консультантів з інформаційної безпеки або розглядати навчання свого персоналу в цій області.

Важливо розуміти, що ефективне забезпечення інформаційної безпеки вимагає комплексного підходу і професійної експертизи.

Швидкість розробки. На думку компанії «Metomic», яка допомагає компаніям захищати конфіденційні дані в програмах, це одним важливим аспектом є брак експертизи в галузі інформаційної безпеки. «Стартапи не дуже дбають про безпеку. Вони просто прагнуть вижити та розвиватися. Тепер, коли перевірка кібербезпеки надходить звідусіль, вони більше не можуть цього уникнути» [3]. Це створює загрозу безпеці, оскільки без належного рівня експертизи може бути важко визначити та вирішити потенційні проблеми. ІТ-стартапи часто працюють в швидкому темпі з метою випуску продукту на ринок якнайшвидше. Прискорена розробка може призвести до ігнорування процесів тестування та безпеки, що створює вразливості в програмному коді та системі в цілому.

Для вирішення цієї проблеми, ІТ-стартапи можуть взяти на озброєння підходи, які дозволяють інтегрувати процеси покращення безпеки на ранніх етапах розвитку продукту. Це включає в себе усвідомлення важливості безпеки від

самого початку, навчання команди стартапу щодо базових практик інформаційної безпеки, а також впровадження автоматизованих інструментів для виявлення потенційних загроз та помилок у коді. Такий підхід допомагає забезпечити більш високий рівень безпеки без значного уповільнення процесу розробки.

Забезпечення довіри клієнтів та інвесторів. Успіх стартапу визначається не лише функціональністю продукту, але й рівнем довіри, який він викликає у клієнтів та інвесторів. Втрати даних або інші інциденти безпеки можуть погіршити репутацію компанії і втратити довіру. «Належні практики безпеки захищають конкурентну перевагу бізнесу, роблячи його більш привабливим для майбутніх інвесторів і клієнтів. Закладання міцного фундаменту з самого початку допоможе їхній безпеці стати більш ефективною та менш затратною в міру розвитку бізнесу» [4]. Зазначена цитата підкреслює критичне значення розробки та впровадження належних практик безпеки у стратегію розвитку стартапу.
Важливість таких практик не тільки у забезпеченні безпеки даних та запобіганні інцидентів, але й у формуванні довіри серед клієнтів та інвесторів. Це впливає на загальну репутацію компанії та її здатність приваблювати додаткове фінансування. Забезпечення високого рівня безпеки з самого початку не тільки сприяє більш ефективному управлінню ресурсами, але й створює стійкий фундамент для майбутнього зростання та успіху стартапу.

Потреба у відповідності правилам і регулюванням. У багатьох сферах, зокрема фінансовій та медичній, існують вимоги до забезпечення безпеки даних та відповідності регулюванням (наприклад, «GDPR» у Європі). У контексті регулювання та відповідності правилам, стартапам у сферах, які мають високі вимоги до безпеки даних, таких як фінансова та медична галузі, необхідно звернути увагу на наступні ключові аспекти:

– зростання кібератак та їх витрат: Кібератаки щорік зростають, а середні витрати на відновлення після порушення даних у США становили 9.44 мільйони доларів у 2022 році. Малі бізнеси часто стають цілями хакерів через недостатні заходи безпеки [5];

– суворість регулювання у фінансовій та медичній галузях: «HIPAA» у

медичній галузі вимагає ретельного захисту електронної охоронної інформації (ePHI) та передбачає ряд важелів контролю для забезпечення безпеки. «GDPR» у Європі вимагає суворого захисту персональних даних, зокрема у фінансовій галузі [5];

– значення управління ризиками та відповідності (GRC): GRC (Governance, Risk, and Compliance) є важливим для забезпечення безпеки інформації та дотримання регулятивних вимог. Управління ризиками та відповідністю включає ідентифікацію потенційних загроз, класифікацію ризиків, оцінку вразливостей та впровадження контролів для їх зменшення [5];

– важливість вибору відповідних рамок безпеки: Залежно від галузі та регіону, стартапи можуть використовувати різні стандарти та рамки безпеки, такі як «NIST», «ISO 27001», «SOC 2», «HIPAA», «GDPR» та інші [5]. Вибір відповідної рамки безпеки може забезпечити не тільки відповідність регулятивним вимогам, але й підвищити довіру з боку клієнтів та інвесторів;

– синергія між управлінням, ризиками та відповідністю правилам і нормам: Управління ризиками, відповідність правилам та галузеві стандарти тісно пов'язані та взаємодоповнюючі [5].

Загалом, для стартапів у високорегульованих галузях, таких як фінансова та медична, активне зосередження на відповідності регулятивним вимогам та розробка ефективних стратегій управління ризиками та безпекою даних є не тільки обов'язковим, але й стратегічно важливим для стійкого розвитку та довіри клієнтів та інвесторів.

Враховуючи ці виклики, інформаційна безпека стає невід'ємною частиною успішного розвитку IT-стартапів. Забезпечення належного рівня безпеки вимагає обізнаності, ефективного використання ресурсів і відповідних стратегій.

1.2 Аналіз сучасних джерел загроз та вразливостей

В інформаційному просторі існує досить значна кількість джерел, що надають велику кількість даних про загрози та вразливості кібербезпеки.

Наприклад – база даних «Common Vulnerabilities and Exposures» (CVE), База

даних вразливостей («VulnDB»), Національна база даних про вразливості (NVD), служать критично важливими інструментами для IT-фахівців та експертів з кібербезпеки[2].

База даних CVE – це загальнодоступний каталог стандартизованих ідентифікаторів відомих загроз безпеки та вразливостей. Вона надає можливість взаємодії з даними у результаті якої можуть бути створені різноманітні інструменти для контролю та втілення практик кібербезпеки.

Подібним чином «VulnDB» пропонує детальну інформацію про вразливості, включаючи власні дослідження та більш детальну категоризацію даних, що може бути особливо цінним для організацій, яким потрібен комплексний аналіз безпеки [3].

Інші платформи, такі як «NVD», надають додаткові рівні аналізу, включаючи оцінки важливості та оцінки впливу на основі загальної системи оцінки вразливостей (CVSS) [3]. Ці оцінки допомагають визначити пріоритетність реагування та заходів безпеки відповідно до потенційного впливу кожної вразливості.

Незважаючи на велику кількість даних про природу, серйозність і деталі вразливостей, ці бази даних не надають прямих кореляцій з конкретними методами або інструментами для їх запобігання або усунення їх наслідків, що додатково підкреслює актуальність проекту, адже його задачею стане наступний крок – створення прямих зв'язків між вразливостями та конкретними методами DevSecOps. Подібне відношення між даними про вразливості та методами їхнього вирішення відсутнє у сучасних джерелах, що робить наш підхід унікальним і цінним.

Наш проект дозволить зрозуміти яким чином стартапи можуть більш цілеспрямовано використовувати DevSecOps інструменти для оптимізації своїх інвестицій в безпеку. Такий інтегрований підхід може значно покращити здатність організацій протистояти сучасним кіберзагрозам, роблячи їх більш гнучкими та стійкими.

Завдання дослідження включає аналіз сучасних методів DevSecOps та їх

адаптацію для використання, розробку моделі машинного навчання для аналізу бази даних «Common Vulnerabilities and Exposures» (CVE) і визначення ефективності цих методів у підвищенні безпеки програмного забезпечення.

Очікується, що в результаті роботи буде визначено необхідні інструменти для раннього виявлення і профілактики потенційних загроз, що можуть значно знизити ризики пов'язані безпекою на ранніх етапах розробки у IT-стартапах.

1.3 Постановка задачі дослідження

Після проведення аналізу проблемної області дослідження та визначення ключових викликів інформаційної безпеки для IT-стартапів, а також аналізу сучасних джерел загроз та вразливостей, ми можемо сформулювали задачі для проведення дослідження. Для досягнення мети дослідження необхідно виконати наступні завдання:

а) теоретичне дослідження існуючих методів DevSecOps:

1) провести огляд існуючих методів DevSecOps, аналізуючи публікації, статті компаній та спеціалістів з предметної області. Це дозволить визначити найактуальніші та ефективні методи впровадження DevSecOps;

2) розглянути приклади успішного впровадження методів DevSecOps у різних компаніях, щоб зрозуміти практичні аспекти їх застосування;

3) вивчити інструменти для впровадження методів DevSecOps. Для кожного методу необхідно визначити відповідні практичні інструменти, що сприяють його реалізації;

б) практичне дослідження загроз інформаційної безпеки на основі моделі даних «Common Vulnerabilities and Exposures» (CVE) та інструментів DevSecOps:

1) розробити модель машинного навчання для класифікації загроз з бази CVE на основі маркування даних. Кожен запис CVE буде позначений відповідним лейблом, який представлятиме інструмент або метод DevSecOps. Модель класифікувати базу CVE за інструментами DevSecOps;

2) створити датасет для майбутньої моделі, виконати маркування даних, тренувати модель та налаштувати її параметри;

3) застосувати модель до реального датасету для визначення актуальності методів DevSecOps у різних сценаріях;

4) проаналізувати результати практичного дослідження, оцінити ефективність моделі та зробити висновки щодо використання різних методів та інструментів DevSecOps;

в) розробка рекомендацій щодо застосування методів та інструментів DevSecOps у IT-стартапах:

1) на основі теоретичного та практичного дослідження створити рекомендації щодо застосування методів DevSecOps у IT-стартапах. Це включатиме визначення найбільш актуальних та ефективних методів та інструментів для забезпечення інформаційної безпеки;

2) сформувати модель загроз для IT-стартапу, що базується на отриманих результатах, яка допоможе IT-стартапам ідентифікувати потенційні загрози та застосовувати відповідні заходи безпеки.

Виконання цих задач дозволить забезпечити систематичний підхід до вивчення передових методів DevSecOps, їх актуальності та шляхів запровадження для підтримки інформаційної безпеки в процесах розробки програмного забезпечення у IT-стартапах

2 ДОСЛІДЖЕННЯ МЕТОДІВ DEVSECOPS В РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Розгляд ключових методів впровадження DevSecOps в розробці програмного забезпечення

Еволюція від DevOps до DevSecOps розкриває важливий шлях у вдосконаленні процесу розробки програмного забезпечення. Почавши як відповідь на потребу в швидшій та більш гнучкій розробці, DevOps спрямував увагу на співпрацю між командами розробників та операцій. Проте, з розвитком DevOps та швидким розгортанням програмного забезпечення, виникла необхідність інтегрувати безпеку безпосередньо в процес розробки.

Ця необхідність в безпеці привела до виникнення DevSecOps – парадигми, де безпека стає вбудованою частиною процесу розробки, а не просто додатковим аспектом. DevSecOps враховує важливість безпеки в усіх аспектах створення програмного забезпечення, надаючи пріоритет захисту даних та систем вже на ранніх етапах розробки [6].

DevSecOps, що є скороченням від «Development, Security, and Operations», представляє собою інтегрований підхід до розробки програмного забезпечення, де безпека є центральною частиною усього процесу розробки та експлуатації[6]. Цей підхід вимагає культурної зміни в організаціях, де команди розробників, операцій та безпеки тісно співпрацюють протягом усього життєвого циклу розробки програмного забезпечення.

Відповідь на сучасні кіберзагрози стає надзвичайно важливою у світі зростаючих загроз інформаційній безпеці. У цьому контексті DevSecOps вважається критично важливим для захисту інформаційних систем. Однією з ключових переваг DevSecOps є можливість виявлення вразливостей на ранніх стадіях розробки, що дозволяє запобігати серйозним проблемам безпеки в майбутньому [7].

Проте, важливо також враховувати, що DevSecOps вимагає зміни культури організацій. У цій новій культурі безпека стає відповідальністю кожного члена

команди розробки, а не тільки спеціалістів з безпеки [7]. Це передбачає розвиток співпраці та взаємодії між всіма учасниками процесу розробки, з метою забезпечення безпеки на всіх етапах розробки програмного забезпечення.

Основні принципи DevSecOps включають:

Зміщення безпеки вліво (Shifting Security Left).

Це основний принцип DevOps і, відповідно, DevSecOps. Це передбачає переміщення процесів – у цьому випадку безпеки – з кінця процесу доставки на початок, відомий як «зміщення ліворуч». Середовища, що використовують DevSecOps розміщують питання безпеки на початку життєвого циклу розробки, вимагаючи взаємодії програмістів і інженерів з безпеки з командою розробки [8].

Команда спільно відповідальна за забезпечення безпеки кожного компонента і конфігурації. Кожен член команди повинен впроваджувати заходи забезпечення безпеки і документувати свої процеси. Перенесення безпеки вліво дозволяє команді виявляти ризики безпеки на ранніх стадіях, що дозволяє негайно виправляти загрози безпеки та полегшує швидкий та безперешкодний процес доставки. Розробники додають безпекові процеси до своїх традиційних процесів розробки.

Тренінги з безпеки.

Інформаційна безпека вимагає поєднання процесів комплаєнс-контролю та інженерії програмного забезпечення. Розробники, інженери програмного забезпечення та фахівці з безпеки повинні тісно співпрацювати з відділом комплаєнс-контролю, щоб всі були ознайомлені з політикою безпеки організації. Усі співробітники повинні періодично проходити навчання, щоб забезпечити їхнє розуміння власних обов'язків [8].

Усі особи, які беруть участь у процесі розробки та доставки програмного забезпечення, повинні бути ознайомлені з основними принципами безпеки додатків, включаючи обізнаність із проектом з безпеки веб-додатків (OWASP), процесами тестування на безпеку та найкращими практиками інженерії програмного забезпечення [10]. Розробники повинні розуміти моделі загроз та оцінки відповідності. Вони повинні знати, як визначати та вимірювати ризики та

вразливості безпеки та застосовувати засоби безпеки.

Культура робочого місця.

Успішне втілення DevSecOps потребує культури робочого місця, яка вітає зміни та серйозно ставиться до безпеки. Керівництво організації повинно сприяти співпраці та комунікації, щоб забезпечити єдиності зусиль у сфері безпеки. Розробники та інженери програмного забезпечення повинні брати на себе відповідальність за процеси безпеки, включені в процес розробки та доставки [8].

Команда що практикує DevSecOps повинна встановити систему, яка включає відповідні практики та технології. Команда повинна мати можливість створювати робоче середовище, яке відповідає її потребам, дозволяючи кожному члену команди відчути інвестиції в успіх проекту.

Спостережливість (observability) та моніторинг.

Збереження інформаційної безпеки вимагає постійного моніторингу та рішень з можливістю спостереження, які надають інформацію про безпеку та допомагають відстежувати ризики розробки [8].

Ефективна стратегія observability повинна включати наступні елементи [8]:

– видимість (Visibility) – здатність бачити процеси розробки та безпеки є важливою для підтримки середовищ DevSecOps. Організації повинні використовувати системи моніторингу для вимірювання операцій, генерування сповіщень та надання інформації про загрози та атаки. Видимість є важливою для забезпечення обліковості на всіх стадіях життєвого циклу проекту;

– відслідковування (Traceability) – організація повинна мати змогу відстежувати конфігурації безпеки та проблеми коду на протязі всього процесу розробки. Це важливо для виконання контролю та дотримання вимог, а також для мінімізації помилок, забезпечення безпеки коду та полегшення виправлення коду;

– ревізійність (Auditability) – організації повинні проводити перевірки для дотримання внутрішніх політик з безпеки та державних регуляцій. Всі засоби безпеки повинні бути піддаються аудиту та мати належну документацію.

Методи DevSecOps – це набір практик та процедур, спрямованих на забезпечення безпеки в розробці та постачанні програмного забезпечення. Ці

методи дозволяють інтегрувати безпеку безпосередньо в процес розробки та підтримувати її на кожному етапі життєвого циклу програми [8]. Вони допомагають у виявленні та вирішенні вразливостей та ризиків безпеки на ранніх стадіях розробки, що дозволяє зменшити витрати та забезпечити вищий рівень безпеки програмного забезпечення.

Моделювання Загроз.

Моделювання загроз визначає можливі сценарії атак, описує потоки чутливих даних, вразливості та можливі варіанти подолання цих загроз [8]. Цей крок допомагає підняти рівень безпеки та покращити знання з безпеки для всіх членів команди.

Тестування Безпеки.

Сканування – це процес аналізу коду, артефактів та робочого програмного забезпечення для виявлення слабкостей в безпеці. Сюди входять як ручні, так і автоматизовані огляди коду, інструменти безпеки додатків, такі як статичне/динамічне тестування безпеки додатків (SAST/DAST) [9], оцінка вразливостей та тестування на проникнення. Цей етап дозволяє розробникам вирішувати вразливості в безпеці та помилки на ранніх етапах життєвого циклу розробки програмного забезпечення.

Аналіз і визначення пріоритетів.

Етап аналізу визначає ризики безпеки, переглядаючи всі дані та показники, зібрані під час тестування безпеки [8]. Ці ризики потім агрегуються у список та ранжуються за можливим впливом на бізнес та ймовірністю експлуатації.

Усунення наслідків.

Після ідентифікації та виявлення вразливостей в безпеці на попередніх етапах, команди вживають заходів для усунення цих вразливостей. Інструменти для постійного тестування та процеси, такі як тестування на проникнення, надають конкретні вказівки щодо вирішення проблем безпеки [8]. Команди можуть потім вирішувати вразливості у порядку їх пріоритету.

Моніторинг.

Моніторинг передбачає відстеження загального стану безпеки додатка з

метою виявлення нових вразливостей або невірних конфігурацій, які можуть виникнути під час його роботи в продакшені. Крім того, моніторинг є критичним для виявлення загроз і порушень безпеки [10]. Коли виявляється загроза або відбувається порушення, цей досвід вивчається для поліпшення процесу DevSecOps та запобігання подібним інцидентам у майбутньому.

2.2 Приклади успішного впровадження методів DevSecOps в організації розробки IT-стартапів

«10x Banking»: Підхід до Моделювання Загроз у DevSecOps.

«10x Banking», компанія, яка спеціалізується на фінансових технологіях, усвідомлює важливість інтеграції безпеки в процес розробки з самого початку. Їх підхід до DevSecOps орієнтований на забезпечення безперервної безпеки в швидкісному середовищі розробки. Одним з ключових елементів їх стратегії є використання загрозового моделювання, що допомагає командам ідентифікувати потенційні загрози безпеці на ранніх етапах розробки.

Застосування карткових ігор з моделювання загроз згідно з моделлю «STRIDE» дозволяє командам у інтерактивній формі виявляти і обговорювати потенційні вразливості та загрози. Ця техніка не тільки сприяє підвищенню обізнаності щодо безпеки серед розробників, але й сприяє креативності та глибшому розумінню безпекових ризиків. «10x Banking» акцентує на тому, що безпека – це не окремий елемент, а інтегрована частина процесу розробки, що вимагає активної участі кожного члена команди [11].

«Datadog»: Інтеграція Безпеки в Процес Розробки.

«Datadog», компанія, яка надає сервіси моніторингу хмарних застосунків, впровадила унікальний підхід до DevSecOps, зосередивши увагу на вбудовуванні безпеки в кожному етапі розробки. Основною стратегією «Datadog» було впровадження спільної відповідальності за безпеку між розробниками та інженерами з безпеки. Вони інтегрували інженерів безпеки безпосередньо в команди розробників, що допомогло підвищити обізнаність про безпеку і забезпечити більш глибоке розуміння безпекових вимог.

«Datadog» розробила власний інструмент для статичного аналізу безпеки та аналізу вразливостей залежностей, який був ефективно інтегрований у їх DevOps пайплайни. Це не тільки підвищило ефективність виявлення вразливостей, але й забезпечило гнучкість у вирішенні безпекових викликів. Завдяки цим зусиллям «Datadog» змогла забезпечити високий рівень безпеки в своїх продуктах, одночасно підтримуючи швидкість і агільність у розробці [11].

«Pivotal»: Культурна Трансформація для DevSecOps.

«Pivotal», компанія, яка надає хмарні рішення та послуги розробки, здійснила культурну трансформацію, щоб адаптуватися до нових парадигм DevSecOps. Вони перейшли від традиційного уявлення про безпеку як процесу-воріт у розробці до створення захисних бар'єрів, які спрямовують розробників до більш безпечних рішень. Цей підхід передбачає тісну співпрацю між розробниками та фахівцями з безпеки, підкреслюючи важливість спільної відповідальності за безпеку.

«Pivotal» акцентує на важливості культурних змін, що включають не тільки технологічні, але й організаційні аспекти. Вони проводять робочі зустрічі та навчання для різних відділів, зосереджуючись на реальностях хмарної безпеки та інтеграції безпеки у DevSecOps. Це дозволило «Pivotal» ефективно адаптуватися до DevSecOps, забезпечуючи безпеку без компромісів у швидкості та інноваційності розробки [11].

Ці приклади демонструють, як різні компанії впроваджують DevSecOps, зосереджуючись на культурній трансформації, співпраці між командами, інтеграції безпеки у процеси розробки, і використанні інноваційних технологій для забезпечення інформаційної безпеки продуктів.

2.3 Практичні інструменти впровадження методів DevSecOps в організації розробки програмного забезпечення

В сучасному цифровому світі, де безпека програмного забезпечення є критично важливою, методи DevSecOps відіграють ключову роль у забезпеченні безперервного інтегрування безпеки в процес розробки. Зокрема, такі методи, як

моделювання загроз, аналіз та визначення пріоритетів, тестування безпеки, усунення наслідків та моніторинг, вимагають використання спеціалізованих інструментів для їх ефективного втілення.

Для кожного методів існують відповідні інструменти. Наприклад, інструменти для створення моделей загроз, інструменти для сканування образів контейнерів, операційних систем, коду, інфраструктури, інструменти тестування на проникнення, захисту мережі, виявлення аномалій та вторгнення, обмеження доступу, зберігання секретів, а також інструменти для зберігання та перегляду логів і метрик. Вибір і використання цих інструментів залежать від різних факторів, таких як наявність ресурсів, обмеження часу, а також їх актуальність та релевантність [12].

Огляд інструментів DevSecOps представлено у таблиці 2.1:

Таблиця 2.1 – Сучасні інструменти та методи DevSecOps (таблиця виконана самостійно)

Метод	Тип Інструменту	Приклад Інструменту	Опис Інструменту
Моделювання Загроз	Інструменти для створення моделі загроз	IriusRisk	IriusRisk – це інструмент для моделювання загроз та оцінки ризиків в програмному проекті. Він дозволяє розробникам та інженерам створювати моделі загроз, ідентифікувати вразливості та оцінювати ризики, пов'язані з програмними продуктами.
Моделювання Загроз	Інструменти для створення моделі загроз	Microsoft Threat Modeling Tool	Microsoft Threat Modeling Tool – це інструмент, розроблений компанією Microsoft, призначений для створення моделей загроз та аналізу вразливостей в інформаційних системах.

Продовження таблиці 2.1

Метод	Тип Інструменту	Приклад Інструменту	Опис Інструменту
Моделювання Загроз	Інструменти для створення моделі загроз	Pytm	Pytm (Python Threat Modeling) – це бібліотека для мови програмування Python, яка допомагає створювати та редагувати моделі загроз та виконувати аналіз безпеки. Вона надає інструменти для розробки та оцінки загроз, визначення вразливостей та прийняття рішень щодо захисту.
Аналіз і визначення пріоритетів	Інструменти зберігання та мониторинг у результатів тестування безпеки	SIEM	SIEM системи (Security Information and Event Management): SIEM системи, такі як LogRhythm, Splunk, Datadog, надають можливості зібрати, аналізувати та моніторити події безпеки з різних джерел для виявлення загроз та визначення їх пріоритетності.
Тестування Безпеки	Інструменти сканування образів контейнерів	Snyk Container	Snyk Container – це частина платформи Snyk, яка спеціалізується на безпеці контейнерів. Цей інструмент призначений для аналізу та забезпечення безпеки образів контейнерів, які використовуються в розробці та експлуатації додатків, що використовують контейнеризацію (наприклад, Docker контейнери).

Продовження таблиці 2.1

Метод	Тип Інструменту	Приклад Інструменту	Опис Інструменту
Тестування Безпеки	Інструменти сканування операційних систем	OpenVAS	OpenVAS (Open Vulnerability Assessment System): OpenVAS є відкритою системою сканування вразливостей, яка надає багато різних сценаріїв сканування для різних операційних систем та програмного забезпечення.
Тестування Безпеки	Інструменти сканування коду	Snyk Code	Snyk Code – це компонент платформи Snyk, який спеціалізується на аналізі безпеки програмного коду. Він призначений для виявлення та виправлення вразливостей та проблем безпеки в програмному коді, який використовується в розробці додатків.
Тестування Безпеки	Інструменти сканування інфраструктур	ScoutSuite	Scout – це інструмент для сканування інфраструктури та виявлення вразливостей. Він допомагає ідентифікувати проблеми у конфігурації серверів та інших компонентів інфраструктури.
Тестування Безпеки	Інструменти сканування коду інфраструктур	Chekhov	Chekhov – це інструмент для сканування коду інфраструктури та перевірки дотримання стандартів безпеки. Він допомагає виявляти вразливості та потенційні проблеми у конфігураціях інфраструктури як коду.

Продовження таблиці 2.1

Метод	Тип Інструменту	Приклад Інструменту	Опис Інструменту
Тестування Безпеки	Інструменти тестування на проникнення	Pentester	Pentester.com – це онлайн-платформа для проведення тестування на проникнення та аналізу безпеки. Вона надає послуги з експертизи безпеки та тестування вразливостей для програмного забезпечення та інфраструктури.
Усунення наслідків	Інструменти захисту мережі	Cloudflare	Cloudflare – це популярна інтегрована платформа з захисту мережі та безпеки в Інтернеті. Вона надає різноманітні послуги та інструменти для захисту веб-додатків та інфраструктури в хмарному середовищі.
Усунення наслідків	Інструменти виявлення аномалій та вторгнення	Snort	Snort – це відкритий сирцевий системний інструмент виявлення і запобігання вторгненням (IDS/IPS), який може аналізувати мережевий трафік і сповіщати про підозрілу активність.
Усунення наслідків	Інструменти зберігання секретів	Vault	Vault – це інструмент для зберігання та керування секретами, такими як паролі, API-ключі та інша конфіденційна інформація. Він допомагає забезпечити безпеку секретів у розподілених середовищах.

Кінець таблиці 2.1

Метод	Тип Інструменту	Приклад Інструменту	Опис Інструменту
Усунення наслідків	Інструменти обмеження доступу	RBAC model	Role-Based Access Control) використовується в таких сервісах, як AWS Identity and Access Management (IAM) та Single Sign-On (SSO), щоб керувати доступом користувачів і ролей до різних ресурсів та послуг в хмарних середовищах та бізнес-додатках. В основі RBAC лежить ідея призначення користувачам або ролям конкретних дозволів або прав доступу, які визначають їхню здатність виконувати певні дії в системі.
Моніторинг	Інструменти зберігання та перегляду логів	ELK Stack	ELK Stack (Elasticsearch, Logstash, Kibana): ELK Stack – це популярний стек інструментів, який включає Elasticsearch для зберігання та пошуку лог-даних, Logstash для збору та обробки лог-даних, а також Kibana для візуалізації та аналізу лог-даних.
Моніторинг	Інструменти зберігання та перегляду метрик	Grafana	Grafana: Grafana – це платформа для візуалізації метричних даних. Вона інтегрується з різними джерелами даних, включаючи Prometheus, та дозволяє створювати графіки та дашборди для аналізу метрик.

Інтеграція в життєвий цикл розробки програмного забезпечення: DevSecOps інтегрує методи безпеки на всіх етапах життєвого циклу розробки програмного забезпечення. Це гарантує раннє та постійне виявлення, оцінку, пріоритетність і пом'якшення вразливостей, що знижує ризики, пов'язані з цими вразливими місцями.

Керування вразливостями в DevSecOps. Процес керування вразливістю в DevSecOps включає виявлення недоліків у коді чи програмному забезпеченні, які можуть надати зловмисникам неавторизований доступ. Це включає як технічні вразливості, такі як помилки, пов'язані з кодом, або неправильні налаштування, так і людські вразливості, як-от помилки в обробці даних або керуванні доступом. Інтеграція з базою даних CVE допомагає ідентифікувати відомі вразливості та їх потенційні експлойти.

Модель зрілості для практик DevSecOps: модель зрілості для DevSecOps включає різні рівні інтеграції та практики безпеки, зокрема керування зображеннями, журналювання, моніторинг, попередження, керування виправленнями, управління платформою та керування змінами. На вищих рівнях зрілості практики DevSecOps включають автоматичне тестування виправлень, інформування розробників додатків про зміни в уразливостях програмного забезпечення, відображені в базі даних CVE, і автоматизацію змін з мінімальним впливом на розробників. Автоматизовані та проактивні заходи безпеки: метою інтеграції DevSecOps з базою даних CVE є проактивне й автоматичне усунення вразливостей безпеки. Це включає використання інструментів для статичного та динамічного тестування безпеки додатків (SAST і DAST) і перевірку нового коду на наявність вразливостей, а CVE служить точкою відліку для відомих вразливостей. Таким чином, інтеграція DevSecOps з базою даних CVE має вирішальне значення для проактивного та комплексного підходу до безпеки при розробці програмного забезпечення. Він гарантує виявлення вразливостей і ефективне керування ними, використовуючи велику базу даних відомих вразливостей, надану CVE, для покращення заходів безпеки протягом усього процесу розробки.

Особливу увагу варто звернути на використання бази «Common Vulnerabilities and Exposures» (CVE) для прогнозування значущості та пріоритетності втілення різних інструментів DevSecOps [13]. На вищих рівнях зрілості практики DevSecOps включають автоматичне тестування виправлень, інформування розробників додатків про зміни в уразливостях програмного забезпечення, відображені в базі даних CVE, і автоматизацію змін з мінімальним впливом на розробників [13].

CVE являє собою публічно доступний каталог вразливостей програмного забезпечення, який може бути використаний для оцінки потенційних ризиків, пов'язаних з конкретними технологіями або компонентами. Інтеграція DevSecOps з базою даних CVE має вирішальне значення для проактивного та комплексного підходу до безпеки при розробці програмного забезпечення. Він гарантує виявлення вразливостей і ефективне керування ними, використовуючи велику базу даних відомих вразливостей, надану CVE, для покращення заходів безпеки протягом усього процесу розробки [13].

Аналіз бази даних CVE відіграє ключову роль у визначенні найактуальніших загроз, що дозволяє IT-стартапам приділяти увагу впровадженню тих інструментів DevSecOps, які найбільше сприятимуть зміцненню їхньої безпеки. Використання даних з CVE та їх аналіз за допомогою машинного навчання забезпечує можливість не тільки виявляти вразливості, але й визначати тренди, які вказують на особливо небезпечні типи атак або найбільш ефективні стратегії їх запобігання.

В рамках дослідження, основними критеріями для порівняння та оцінки інструментів DevSecOps будуть тренди розвитку відомих вразливостей за роками та кількістю згадувань, які класифікуються на основі категорій. Кожна категорія відповідатиме конкретним методам DevSecOps, спрямованим на боротьбу з цими вразливостями. Цей підхід дозволяє не тільки виявляти і ранжувати вразливості за їх актуальністю та розповсюдженістю, але й аналізувати ефективність застосування певних методів DevSecOps для кожної категорії вразливостей.

Застосування машинного навчання до даних CVE для аналізу дозволить

систематично обробити інформацію про тренди появи нових вразливостей та їх категоризації. Такий підхід сприятиме формуванню ієрархічної моделі прийняття рішень у контексті інвестування ресурсів у різні інструменти DevSecOps.

3 ДОСЛІДЖЕННЯ ЗАГРОЗ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ НА ОСНОВІ МОДЕЛІ ДАНИХ «COMMON VULNERABILITIES AND EXPOSURES» ТА ІНСТРУМЕНТІВ DEVSECOPS

3.1 Обробка вихідних даних та створення датасету для моделі

У цьому проекті ми створимо модель для аналізу та кластеризації вразливостей, використовуючи датасет «Common Vulnerabilities and Exposures» (CVE) [13]. Цей датасет є одним із ключових ресурсів у галузі кібербезпеки, оскільки він надає детальну інформацію про відомі вразливості в програмному забезпеченні.

Мета цього аналізу полягає у виявленні та класифікації різних типів вразливостей, що дозволить нам ідентифікувати, які інструменти DevSecOps є найбільш актуальними для конкретних сценаріїв ризику. Такий підхід допоможе краще зрозуміти та пріоритизувати заходи безпеки, необхідні для захисту інформаційних систем.

Для роботи з датасетом ми обрали метод Gradient Boosting Machines (GBM) що є потужним методом машинного навчання, який використовує принципи підсилення для створення сильного класифікатора через послідовне комбінування слабких класифікаторів [14]. Також розглядалися варіанти, як k-means, Agglomerative Hierarchical Clustering [15], але GBM був обраний через його високу точність і здатність працювати з нелінійними даними, а основна перевага GBM полягає в його здатності ефективної оптимізації на різних типах даних та вирішення складних, нелінійних задачі класифікації і регресії.

Робота за датасетом для досягнення мети виконується у декілька етапів, таких як: попередня обробка даних, векторизація тексту, створення нових атрибутів, тренування моделі, перевірка ефективності моделі.

Попередня обробка даних це етап, що включає взаємодію з датасетом з офіційного джерела Програми CVE («Common Vulnerabilities and Exposures») [13]. Завантажений датасет у необробленому вигляді має структуру директорій, що містять JSON файли, кожен з яких описує конкретну вразливість.

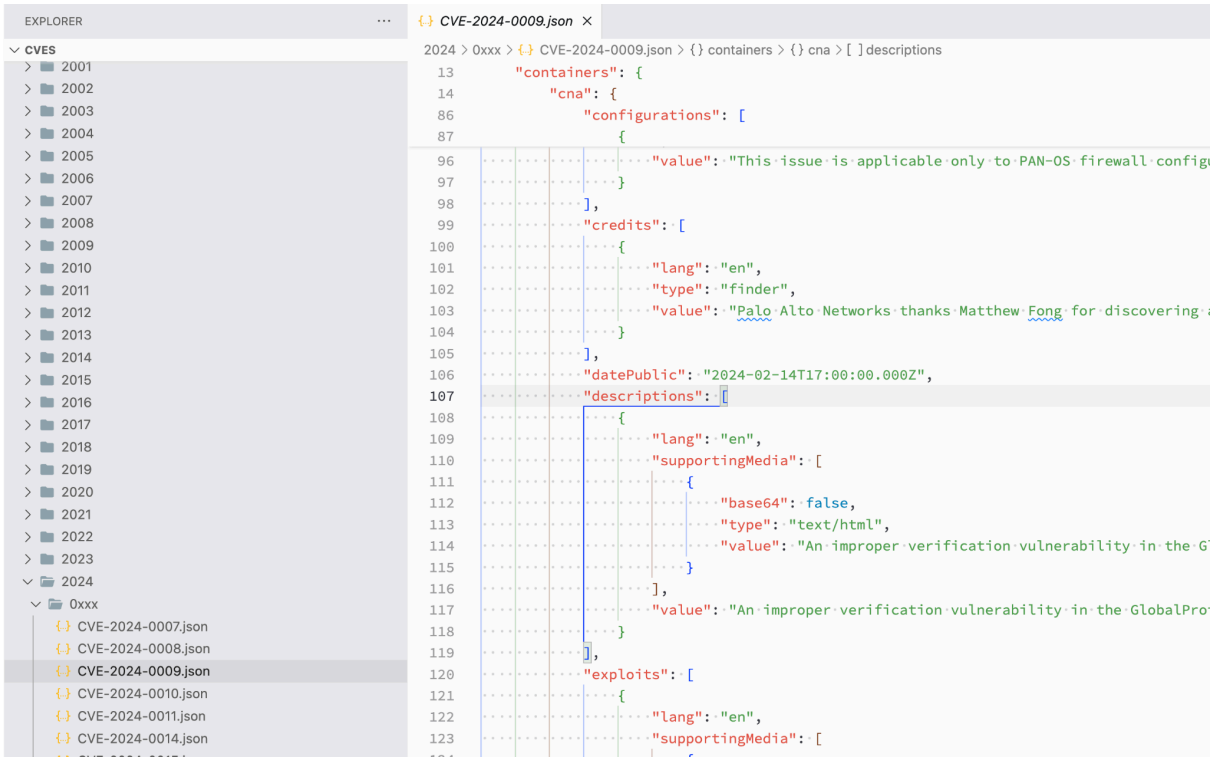

Рисунок 3.1 – Структура датасету у необробленому вигляді (рисунок виконано самостійно)

Ці файли містять безліч атрибутів з якими можна працювати для втілення різних цілей. Для нашого проекту, із необробленого датасету потрібно сформувати більш стислий набір даних, що буде використано у подальшому для взаємодії з моделлю GBM. Вхідні дані, необхідні нам для роботи будуть мати наступний вигляд:

Таблиця 3.1 – Шаблон вхідних даних (таблиця виконана самостійно)

CVE	Description	Severity
CVE-2024-0007	A cross-site scripting (XSS) vulnerability in Palo Alto Networks	MEDIUM

Новий, більш деталізований набір даних буде включати наступні поля:

– «CVE» – унікальний ідентифікатор вразливості;

– «Description» – текстовий опис вразливості;

– «Severity» – негативний вплив вразливості на функціональність з якою вона пов'язана (HIGH, MEDIUM, LOW).

Створення датасету є нетривіальною задачею, через складну структуру директорії та атрибутів JSON файлів, що містять опис вразливостей. Тому, для його генерації створено і застосовано скрипт на основі JavaScript, суть якого полягає у пошуку JSON файлів, фільтрації атрибутів для створення файлу формату CSV (Comma-Separated Values) що містить лише необхідні дані – CVE, Description, Severity (див. додаток Д.1):

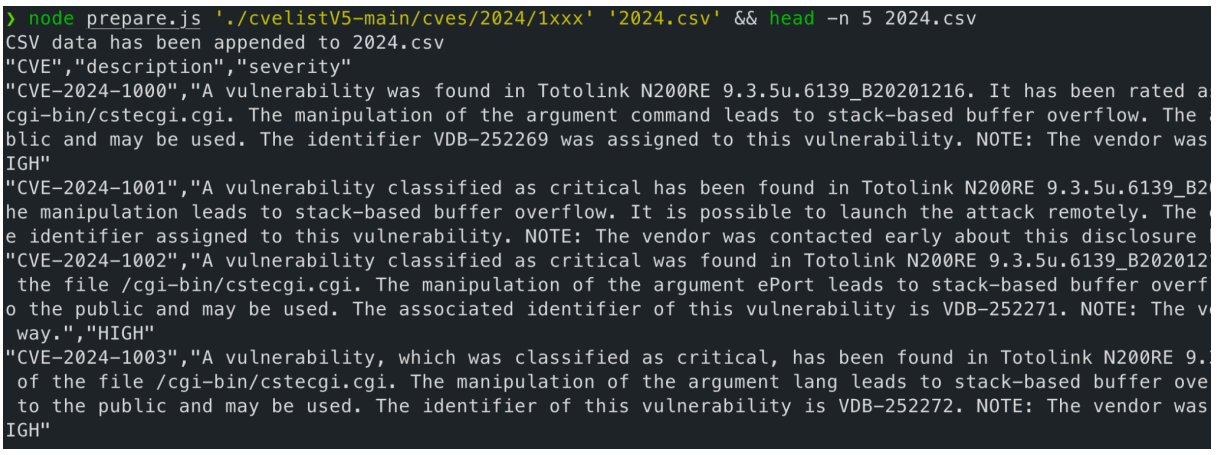

Рисунок 3.2 – Результат обробки початкового датасету за допомогою допоміжного скрипта на JavaScript (рисунок виконано самостійно)

Таким чином ми сформували вхідний датасет для подальшої роботи з моделлю. Він сформований на основі файлів, що містять відомості за останні 5 років 2020-2024.

3.2 Підготовка датасету для тренування та моделювання

База даних «Common Vulnerabilities and Exposures» одним з ключових ресурсів у галузі кібербезпеки, що пропонує стандартизований список публічно оприлюднених вразливостей і ризиків кібербезпеки.

Компанії, що займаються розробкою рішень безпеки, широко використовують базу даних CVE для позначення та класифікації даних, що стосуються загроз безпеці. Наприклад, інструменти керування вразливостями, такі як VulnDB, інтегрують дані CVE для навчання алгоритмів для розпізнавання та прогнозування вразливостей, щоб збагатити свої бази даних детальними описами та оцінками впливу різних слабких місць безпеки[18].

У контексті машинного навчання база даних CVE є важливою для навчання алгоритмів для розпізнавання та прогнозування вразливостей. Маркуючи набори даних з ідентифікаторами CVE, моделі машинного навчання можуть вивчати шаблони та маркери, пов’язані з конкретними вразливими місцями.

Ця практика дає змогу інструментам аналітики прогнозувати потенційні порушення безпеки на основі історичних даних, значно покращуючи механізми проактивного захисту. Крім того, використання даних CVE допомагає в розробці автоматизованих систем безпеки, які можуть ідентифікувати, класифікувати та реагувати на загрози з мінімальним втручанням людини, таким чином оптимізуючи рівень безпеки організацій [18].

Процес маркування даних (Labeling) є ключовим етапом в підготовці датасетів для машинного навчання, особливо в задачах класифікації. Розмітка даних полягає в анотації датасету мітками (лейблами), які описують відповідний контент або класифікаційну категорію для кожного елементу даних. Ці мітки використовуються алгоритмами машинного навчання для навчання на прикладах та формування моделей, які можуть передбачати мітки для нових, нерозмічених даних [16].

Відповідно до плану нашого проекту, мітками даних будуть виступати практичні засоби втілення DevSecOps. Згідно з описом (description) кожної вразливості, необхідно визначити категорію до якої відноситься кожна CVE, що міститься у файлі вхідних даних [17]:

– інструменти для створення моделі загроз;

– інструменти зберігання та моніторингу результатів тестування безпеки;

– інструменти сканування образів контейнерів;

– інструменти сканування операційних систем;

– інструменти сканування коду;

– інструменти сканування інфраструктури;

– інструменти сканування коду інфраструктури;

– інструменти тестування на проникнення;

– інструменти захисту мережі;

– інструменти виявлення аномалій та вторгнення;

– інструменти обмеження доступу;

– інструменти зберігання секретів;

– інструменти зберігання та перегляду логів;

– інструменти зберігання та перегляду метрик.

Після маркування, датасет буде мати додаткове поле «tool», що являтиме собою основу класифікації, див табл 3.2:

Таблиця 3.2 – Шаблон даних з мітками (таблиця виконана самостійно)

CVE	Description	Severity	Tool
CVE-2024-0007	A cross-site scripting (XSS) vulnerability in Palo Alto Networks	MEDIUM	Code scanning tools
CVE-2024-0008	Web sessions in the management interface ..	MEDIUM	Network protection tools
CVE-2024-0009	An improper verification vulnerability...	MEDIUM	Access restriction tools

Процес маркування даних є досить повільним та складним, адже для його виконання необхідно вручну обробити багато записів про вразливості. На жаль, автоматизація такого процесу є складним завданням, і одним з небагатьох способів пришвидшення може виступати взаємодія з великими мовними моделями, таких як GPT (Generative pre-trained transformer).

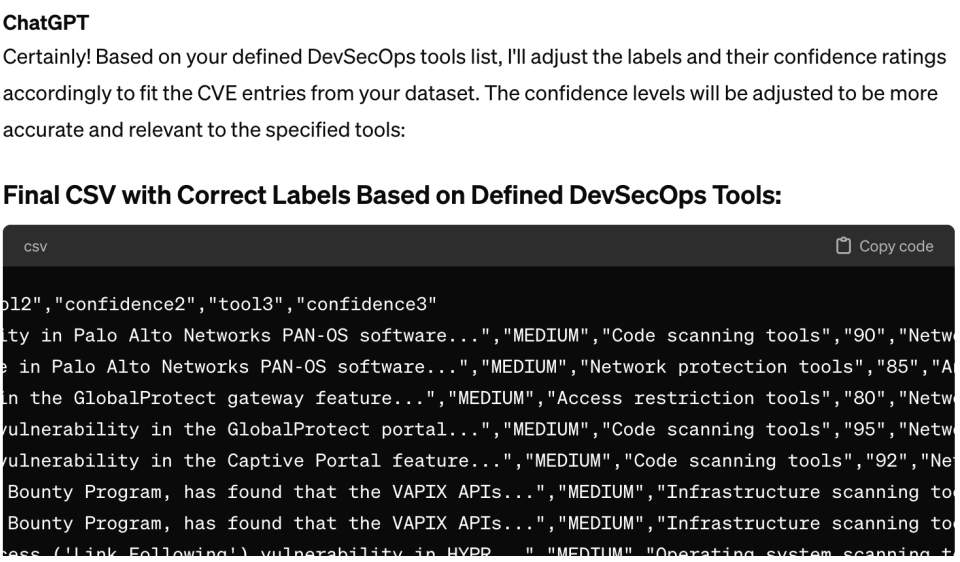

Рисунок 3.3 – Взаємодія з моделлю GPT-4 через ChatGPT для прискорення маркування датасету (рисунок виконано самостійно)

Однак, цей метод не може вважатися абсолютно надійним, він повинен використовуватися лише у якості асистування при ручній обробці датасету.

3.3 Тренування та перевірка ефективності моделі

Модель Gradient Boosting Machines підходить для нашої задачі, оскільки вона може обробляти великі обсяги даних і виявляти складні шаблони в даних опису вразливостей CVE, які можуть бути не очевидними для більш простих моделей. Gradient Boosting Machines може бути описаний наступною основною формулою (3.1), яка показує процес підсилення [14]:

$$F(x) = F_0(x) + \sum_{m=1}^{M} \gamma_m h_m(x) \tag{3.1}$$

де F(x) – прогнозована модель,

$F_0(x)$ – початкова модель,

h_m – слабкий класифікатор (також відомий як базовий учень),

γ_m – коефіцієнт, що визначає внесок кожного слабкого класифікатора в кінцеву модель,

M – загальна кількість слабких класифікаторів.

Для реалізації моделі ми вибрали інтегрований підхід, який включає використання Python, Jupyter Notebook, бібліотеки «pandas», «Scikit-learn» та алгоритму Gradient Boosting Machines (GBM). Код для тренування моделі та оптимізації її параметрів у машинному навчанні включає кілька ключових етапів, кожен з яких відіграє важливу роль у процесі підготовки, тренування та валідації моделі (див. додаток Д.3).

Нижче описано кожен з цих етапів, що втілює необхідні дії для тренування та валідації.

Спочатку відбувається завантаження необхідних бібліотек. Далі дані завантажуються з CSV файлу. Відсутні значення у текстових колонках, замінюються для запобігання проблемам у векторизації тексту.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split, GridSearchCV,
StratifiedKFold

# Load the labeled data
labeled_data = pd.read_csv('labeled-dataset.csv')

# Check for missing values in text columns and fill with 'Missing'
text_columns = labeled_data.select_dtypes(include=['object']).columns
labeled_data[text_columns]                                          =
labeled_data[text_columns].fillna('Missing')
```

Поля опису вразливості та мітки для класифікації переформатуються у числовий формат, що необхідно для роботи алгоритмів машинного навчання:

```
# Extract features and labels
X = labeled_data['description']
y = labeled_data['tool']

# Encode the 'tool' labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

Текстові дані конвертуються в числовий формат через TF-IDF векторизацію, щоб їх могли обробляти алгоритми машинного навчання. Встановлюється система «Pipeline», яка забезпечує автоматичне виконання. Створюється механізм «Pipeline», який дозволяє автоматично виконувати ряд послідовних кроків обробки даних і застосування моделі машинного навчання. Створення pipeline є дуже корисним при розробці комплексних моделей машинного навчання, оскільки воно допомагає уникнути помилок, пов'язаних із «забрудненням» тестових даних під час тренування моделі, та спрощує процес валідації моделі:

```
# Set up a pipeline with TF-IDF Vectorizer and Gradient Boosting Classifier
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer(stop_words='english')),
    ('clf', GradientBoostingClassifier(random_state=0))
])
```

Для автоматичного тестування різних комбінацій параметрів за допомогою стратифікованої кросс-валідації використовується «GridSearchCV» що працює з початковою сіткою параметрів. Це дозволяє вибрати найкращі параметри на основі точності моделі:

```
#Define a grid of parameters to search (including both the vectorizer and the classifier)

param_grid = {
    'tfidf__max_features': [500, 1000, None],
    'tfidf__ngram_range': [(1,1), (1,2)],
    'clf__n_estimators': [100, 200],
    'clf__learning_rate': [0.05, 0.1, 0.2],
    'clf__max_depth': [3, 5, 7]
}

# Create a StratifiedKFold object to use for cross-validation
stratified_k_fold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Create a GridSearchCV object to evaluate the pipeline using the parameter grid and StratifiedKFold
grid_search = GridSearchCV(pipeline, param_grid, cv=stratified_k_fold, scoring='accuracy', verbose=1)
```

Дані розділяються на тренувальний і тестовий набори. Модель тренується на тренувальному наборі та відбувається оцінка точності найкращої моделі на тестовому наборі даних. У результаті отримаємо набір найкращих параметрів, які були визначені під час пошуку:

```
# Split data into training and testing set
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded,
test_size=0.2, random_state=42)

# Fit the grid search to the data
grid_search.fit(X_train, y_train)

# Evaluate the best grid search pipeline on the test set
predictions = grid_search.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print(f"Best model accuracy: {accuracy:.2%}")
print("Best parameters found:", grid_search.best_params_)
```

У ході роботи із тренування та валідації моделі, нашою метою є досягнення ефективності на рівні не меншому за 70%. Для цього виконується значна кількість ітерацій покращення вхідного датасету – збільшення кількості даних та покращення якості маркування:

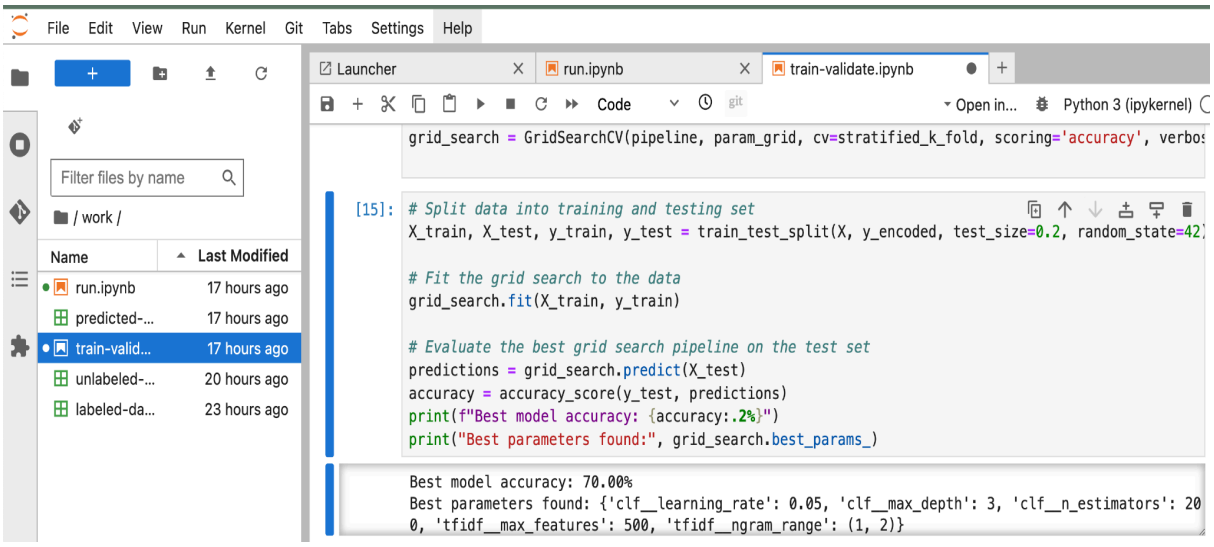

Рисунок 3.4 – проміжний результат тренування та валідації моделі (рисунок виконано самостійно)

Після досягнення цільового показника ефективності моделі машинного

навчання на рівні не менше 70%, можна вважати, що модель належним чином розуміє і відтворює залежності у даних, що були їй представлені.

Такий рівень точності часто вважається достатнім для багатьох практичних застосувань, особливо в умовах, де деяка помилка є прийнятною. Досягнення цього порогу є сигналом до наступних дій – безпосереднє застосування моделі на великому датасеті вразливостей.

3.4 Застосування моделі для визначення актуальності методів DevSecOps

Після досягнення бажаного рівня ефективності, та визначення параметрів моделі, необхідно оновити код механізму «Pipeline» вказавши найкращі параметри для роботи.

```
# Create a pipeline using the best parameters found from the grid search
pipeline = Pipeline([
     ('tfidf', TfidfVectorizer(stop_words='english', max_features=500, ngram_range=(1, 2))),
              ('clf',   GradientBoostingClassifier(learning_rate=0.05, max_depth=3, n_estimators=200, random_state=0))
])
```

Останнім кроком, створимо код для застосування нашої моделі. Це включає створення нового файлу на основі немаркованого датасету, до якого додається прогнозоване значення атрибуту «tool»:

```
# Predict 'tool' labels for the unlabeled dataset
unlabeled_data['tool'] = label_encoder.inverse_transform(pipeline.predict(unlabeled_data['description']))

# Save the new dataframe with predicted labels
unlabeled_data.to_csv('predicted-unlabeled-dataset.csv', index=False)
print("Predictions added and saved to 'unlabeled-dataset-with-predictions.csv'")
```

Таким чином, програмний код (див. додаток Д.2) виконує наступні операції:

– завантаження тренувального датасету, що містить правильні мітки з файлу «labeled-dataset.csv»;

– виконання механізму «Pipeline», що використовує тренувальний датасет для «передбачення» та подальшого маркування вихідного датасету «unlabeled-dataset.csv»;

– створюється кінцевий результат у вигляді файлу «predicted-unlabeled-dataset.csv», що містить копію «unlabeled-dataset.csv» але з маркуванням кожної вразливості відповідно до умов класифікації.

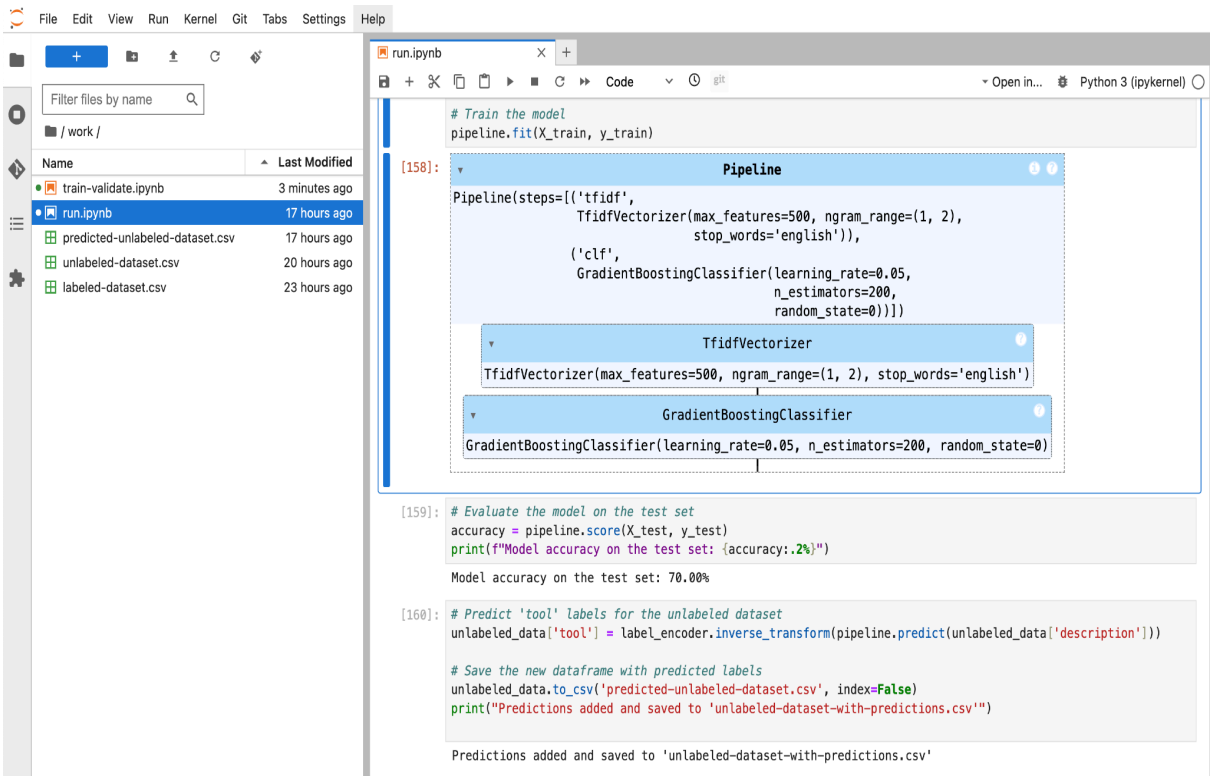

Рисунок 3.5 – результат застосування моделі для генерації кінцевих даних (рисунок виконано самостійно)

Створені дані на основі вразливостей із датасету «Common Vulnerabilities and Exposures» містять відомості щодо актуальності застосування методів та інструментів DevSecOps, та можуть бути використані для подальшого аналізу.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

4.1 Аналіз результатів теоретичного дослідження

Під час теоретичного дослідження було визначено існуючі методи DevSecOps, які використовуються в розробці програмного забезпечення, вивчено їх переваги та недоліків, а також розглянуто практичні приклади успішного впровадження DevSecOps у різних організаціях.

Основні результати дослідження можна підсумувати таким чином:

– ідентифікація методів DevSecOps: було виявлено кілька ключових методів DevSecOps, що використовуються в сучасній розробці програмного забезпечення, включаючи зміщення безпеки вліво (shift left), безперервне тестування безпеки (continuous security testing), інтеграцію автоматизованих інструментів для виявлення та усунення вразливостей, а також моніторинг і аналіз безпеки у режимі реального часу.

– аналіз переваг та недоліків методів DevSecOps: кожен з розглянутих методів має свої переваги та недоліки. Наприклад, зміщення безпеки вліво дозволяє виявляти вразливості на ранніх етапах розробки, що знижує вартість їх усунення. Однак, цей підхід вимагає високого рівня обізнаності та навчання з боку розробників. Безперервне тестування безпеки забезпечує постійну перевірку коду на вразливості, але може бути ресурсомістким і потребує інтеграції з існуючими процесами розробки.

– практичні приклади впровадження DevSecOps: у ході дослідження було розглянуто кілька прикладів успішного впровадження DevSecOps у відомих компаніях. Так, компанія «Pivotal» змогла знизити кількість вразливостей на 40% завдяки інтеграції автоматизованих інструментів для сканування коду та тестування безпеки.

– інший приклад показав, що компанія Datadog» успішно впровадила моніторинг у режимі реального часу, що дозволило вчасно реагувати на нові загрози і зменшити час на їх усунення.

Таким чином, теоретичне дослідження показало, що впровадження методів

DevSecOps у процес розробки програмного забезпечення IT-стартапів може значно підвищити рівень інформаційної безпеки, а практичні приклади підтверджують ефективність цих підходів, демонструючи позитивні тенденції покращення рівня інформаційно безпеки, такі як зниження кількості вразливостей та покращення реакції на нові загрози.

4.2 Аналіз результатів моделювання загроз інформаційної безпеки

У результаті практичного дослідження, було створено набір даних, що показують зв'язок методів та інструментів DevSecOps та загальновідомих вразливостей. Створені дані необхідно проаналізувати для отримання більш глибокого розуміння про вразливості, тренди за роками, кількістю і важливістю та оцінити можливий вплив і зв'язок з DevSecOps.

Спочатку проведемо загальний огляд за трендами, які можна спостерігати на основі зібраних даних. Створимо зведену таблицю за останні 5 років, що містить дані про кількість зареєстрованих вразливостей та їх вагомість, та візуалізуємо дані для подальшого аналізу:

Таблиця 4.1 – загальна кількість вразливостей за роками (таблиця виконана самостійно)

	YEAR				
SEVERITY	2020	2021	2022	2023	2024
CRITICAL	572	756	896	1323	318
HIGH	4422	7089	7896	10107	2658
MEDIUM	2843	3831	5039	8215	2672
LOW	4793	7535	8880	11728	2909

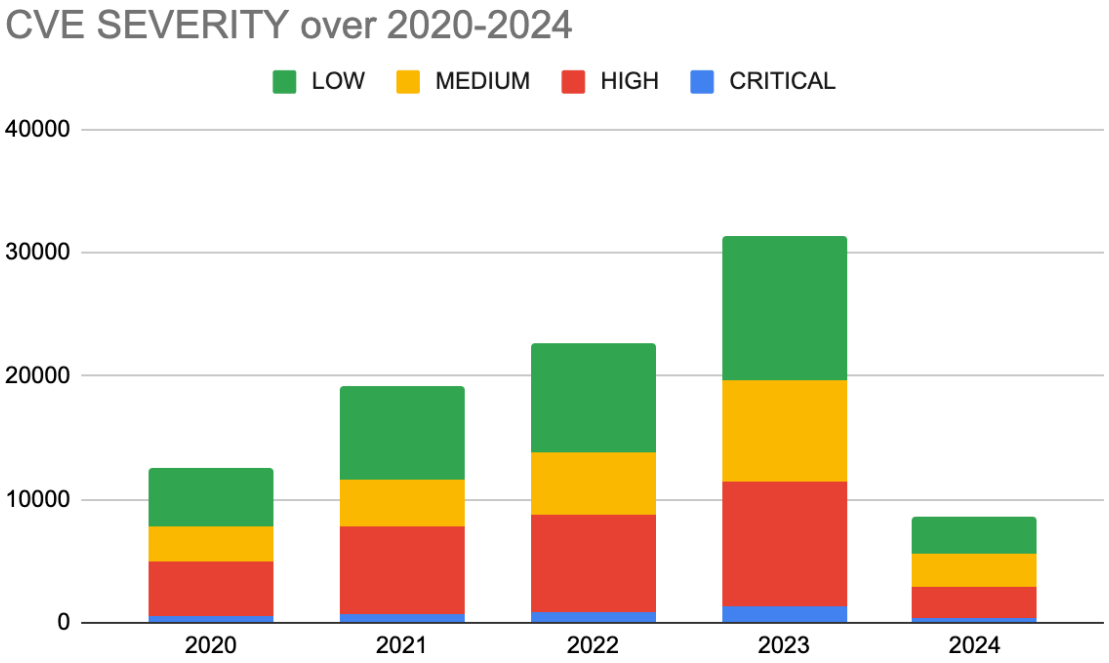

Рисунок 4.1 – Загальна кількість вразливостей за роками (рисунок виконано самостійно)

Дані про вразливості з датасету «Common Vulnerabilities and Exposures» з 2020 по 2024 рік ілюструють чітку тенденцію на ландшафті кібербезпеки: кількість вразливостей з часом стає більшою та серйозною з точки зору їх впливу, а також особлива ескалація спостерігається в категоріях CRITICAL та HIGH. Ця закономірність може свідчити про те, що, оскільки цифрові системи стають більш складними та широко інтегрованими, вони також стають більш вразливими до складних атак. Також, послідовне зростання в усіх категоріях вагомості до 2023 року може свідчити про кращі технології виявлення та більш комплексні стандарти звітності.

Для більш детального огляду вразливостей, та визначення інструментів і методів DevSecOps, що мають прямий зв'язок із цими вразливостями, створимо набір даних на основі датасету створеного нашою моделлю. Для цього використаємо бібліотеку «pandas», що дозволить нам створити таблицю з підрахунками кількості та розподілом вразливостей згідно з їх вагою та відповідними інструментами DevSecOps. проведемо сортування результатів за

загальною кількістю вразливостей, щоб отримати попередній рейтинг інструментів і методів DevSecOps:

Таблиця 4.2 – Розподіл вразливостей за критеріями класифікації (таблиця виконана самостійно)

DevSecOps Tools/methods	CVE Critical	CVE High	CVE Medium	CVE Low	CVE TOTAL
Secrets and credentials storage tools	514	4406	2081	2819	9820
Code scanning tools	508	3849	699	4572	9628
Network protection tools	479	3272	3254	1127	8132
Log storage and viewing tools	439	3032	2395	2148	8014
Tools for storing and viewing metrics	395	3019	1939	1528	6881
Penetration testing tools	361	2606	1859	1912	6738
Access restriction tools	336	2544	479	3145	6504
Anomaly and intrusion detection tools	235	2038	381	3667	6321
Threat modeling tools	219	1736	1512	2660	6127
Security test results storage and monitoring tools (SIEM)	163	1681	694	3477	6015
Container image scanning tools	110	1245	1637	2423	5415
Operating system scanning tools	91	1167	139	3989	5386
Infrastructure scanning tools	90	1070	2447	1762	5369
Infrastructure code scanning tools	82	350	3084	616	4132

Візуалізуємо результат у діаграму для наглядного аналізу даних:

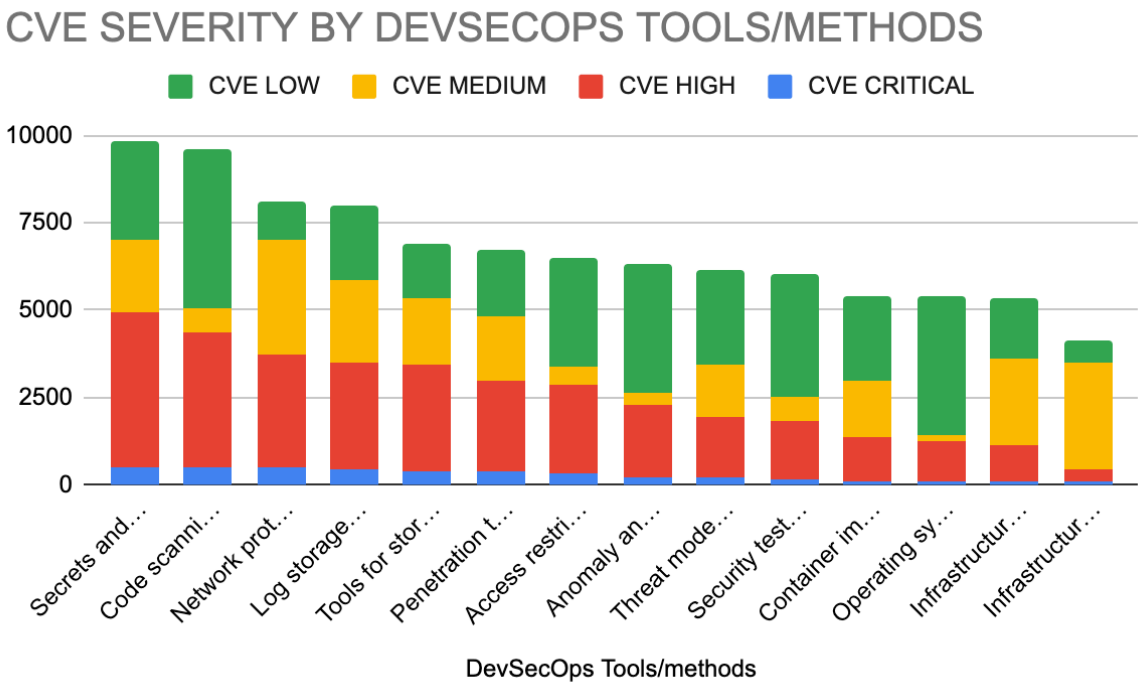

Рисунок 4.2 – Розподіл інструментів і методів DevSecOps за вагою вразливостей (рисунок виконано самостійно)

Розподіл даних вказує на те, що при використанні методів DevSecOps особливу увагу слід зосередити на інструментах для зберігання секретів, сканування коду, захисту мережі, та моніторингу – для зберігання та перегляду логів. Цікавим фактом є те що, дані інструменти займають найбільшу частку відносно вразливостей з рівнем CRITICAL та HIGH, що свідчить про значну актуальність та необхідність їх застосування особливо на ранніх етапах розвитку проекту. Такий розподіл допомагає встановити пріоритети в інвестиціях у інструменти безпеки та стратегії DevSecOps.

Можемо констатувати, що згідно аналізу, дані інструменти DevSecOps мають потенціал ефективного зменшення ризиків або навіть повного блокування вразливостей. Впровадивши їх своєчасно, організації можуть оптимізувати свої процеси розробки програмного забезпечення. Наприклад – сканування коду на ранніх стадіях розробки значно знижує ймовірність виникнення XSS або SQL

Injection, а впровадження механізмів зберігання секретів дозволяє попередити втрату важливих даних.

Також за у результаті створення такого набору даних стає очевидним те, які аспекти DevSecOps потребують більш глибокого розуміння від команд розробників. На базі цих даних можуть бути розроблені спеціалізовані тренінги і навчальні курси, що зосереджені на найбільш актуальних інструментах та методах. Це дозволить не тільки підвищити безпеку майбутніх продуктів, але й оптимізувати витрати на кібербезпеку, що є особливо важливим для середовищ IT-стартапів де завжди наявні обмеження бюджету.

Інформація про ефективність конкретних інструментів DevSecOps може сприяти кращому розумінню і співпраці між відділами розробки, тестування і контролю безпеки. Це, в свою чергу, веде до більш організованого та ефективного процесу виробництва програмного забезпечення.

Результати підкреслюють необхідність інтеграції нових технологій і інструментів, які можуть допомогти у вирішенні потенційно актуальних проблем пов'язаних з безпекою. Це стимулює інновації та постійне вдосконалення інструментів та методів DevSecOps.

5 РЕКОМЕНДАЦІЇ ЩОДО ЗАСТОСУВАННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

5.1 Рекомендації щодо застосування методів DevSecOps у IT-стартапах

Для IT-стартапів початкові етапи розробки є дуже важливими, адже вони змушені постійно визначити пріоритети напрямку своїх зусиль.

Коли ресурси часто обмежені, як з точки зору бюджету, так і робочої сили – пріоритетність заходів кібербезпеки має бути орієнтована на найбільш ефективні та актуальні [6].

Зосередившись на найважливіших інструментах DevSecOps, що виявлені у ході аналізу, стартапи можуть ефективно керувати своїми ресурсами, щоб спочатку зменшити найбільш актуальні та реальні ризики.

Наступні рекомендації визначають конкретні інструменти DevSecOps, пов’язані з найбільшою вразливістю згідно з даними CVE, забезпечуючи цілеспрямований підхід до зміцнення стану безпеки стартапу.

а) інструменти зберігання секретів і облікових даних;

1) пропозиція інструменту: HashiCorp Vault

2) впровадження: стартапам слід інтегрувати Vault на ранніх стадіях життєвого циклу розробки, щоб безпечно керувати секретами та обліковими даними. Це допомагає централізувати зберігання конфіденційної інформації та отримати доступ до неї на основі політик, що може значно зменшити ризик, який підкреслюється великою кількістю пов’язаних CVE.

б) інструменти сканування коду;

1) пропозиція інструменту: Snyk

2) впровадження: запровадити Snyk для автоматичного сканування сховищ коду на наявність вразливостей на усіх етапах розробки. Snyk забезпечує зворотній зв’язок у реальному часі та може бути інтегрований у CI/CD, що допомагає ефективно потенційно зменшити ризик критичних, важливих та середніх за вагою вразливостей згаданих у CVE.

в) засоби захисту мережі;

1) пропозиція інструменту: Palo Alto Networks

2) впровадження: застосувати рішення захисту мережі від Palo Alto для моніторингу та захисту трафіку даних у мережі, що має відношення до продукту та стартапу. Рання інтеграція подібних інструментів може допомогти зменшити ризики, пов’язані з великою кількістю середніх за вагою вразливостей CVE, що постійно з'являються у базі даних, і забезпечити надійну безпеку мережі.

г) інструменти зберігання та перегляду логів;

1) пропозиція інструменту: Splunk

2) впровадження: Splunk можна використовувати для керування та аналізу журналів(логів) у реальному часі. Враховуючи високу кількість CVE на всіх рівнях важливості, стартапам слід налаштувати моніторинг журналів, щоб швидко виявляти потенційні загрози безпецки та реагувати на них.

д) інструменти для зберігання та перегляду метрик;

1) пропозиція інструменту: Grafana

2) впровадження: Grafana слугує інструментом для комплексної візуалізації та моніторингу метрик. Впровадження допоможе вчасно виявити аномальні сценарії поведінки у сервісах та інфраструктурі а також за необхідності зрозуміти стан безпеки шляхом візуалізації даних про вразливості.

е) інструменти тестування на проникнення;

1) пропозиція інструменту: pentester.com

2) впровадження: регулярне планування тестування на проникнення за допомогою pentester.com для раннього виявлення вразливостей, до того як продукт потрапить до користувачів. Цей проактивний підхід має важливе значення, особливо з огляду на значну кількість пов’язаних із цим вразливостей CVE високого та середнього ступеня важливості.

Такий підхід не тільки допоможе вчасно керувати ризиками, але й забезпечить відповідність сучасним стандартам безпеки і підвищить загальну готовність стартапу з точки зору кібербезпеки, що робить його практичною та необхідною стратегією для нових компаній у технологічній галузі.

5.2 Модель загроз для IT-стартапу на основі загальновідомих вразливостей та методів DevSecOps

Під час дослідження ми визначили ключові інструменти, які необхідно використовувати, і відштовхуючись від них, можемо ідентифікувати відповідні загрози та створити модель загроз, що може бути використана IT-стартапом незалежно від роду діяльності та конкретних компонентів системи. Для цього, застосуємо принципи методологій та стандарти, такі як STRIDE [20], OWASP Threat Modeling, що мають у своїй основі ідентифікацію загроз, оцінку ризиків, розробку контрзаходів, моніторинг та постійне вдосконалення [21].

У таблиці 4.3 представлена модель загроз для IT-стартапу, що побудована на основі дослідження з доцільності використання методів та інструментів DevSecOps. Така модель може бути використана стартапом незалежно від конкретних компонентів системи, що використовуються у конкретних проектах, адже дані загрози та інструменти їх попередження або усунення мають широкий спектр впливу і не зосереджені на спеціалізованих технологіях.

Таблиця 4.3 – Модель загроз на основі методів DevSecOps (таблиця виконана самостійно)

Загроза	Вразливість	Рейтинг та строки втілення	Інструменти та методи для усунення	Інтеграція
Витік секретів і облікових даних	Недостатній захист конфіденційної інформації	Високий, негайно	Secrets and credentials storage tools (Vault)	Централізоване сховище для секретів, автоматична ротація ключів

Продовження таблиці 4.3

Загроза	Вразливість	Рейтинг та строки втілення	Інструменти та методи для усунення	Інтеграція
Вразливості в програмному коді	Недостатнє очищення вхідних даних, уразливості типу XSS, SQLi	Високий, протягом перших тижнів	Code scanning tools (Snyk Code)	Автоматичне сканування коду під час кожного коміту в репозиторій
Мережеві атаки (DDoS, Man-in-the-Middle)	Незахищені мережеві з'єднання	Високий, протягом перших місяців	Network protection tools (Cloudflare)	Використання мережевих фільтрів, VPN, системи запобігання DDoS
Недостатній моніторинг і аудит	Недостатнє логування подій	Високий, протягом перших тижнів	Log storage and viewing tools (ELK Stack)	Центральне сховище логів з можливістю їх перегляду і аналізу
Відсутність відстеження стану системи	Події, що загрожують функціям системи	Високий, протягом перших місяців	Tools for storing and viewing metrics (Grafana)	Налаштування дашбордів для моніторингу критичних метрик
Недостатнє тестування безпеки	Приховані вразливості в системі	Високий, після 2-3 перших місяців	Penetration testing tools (Pentester)	Регулярне проведення пенетраційних тестів

Продовження таблиці 4.3

Загроза	Вразливість	Рейтинг та строки втілення	Інструменти та методи для усунення	Інтеграція
Незаконний доступ до системи	Недостатній контроль доступу	Високий, протягом перших місяців	Access restriction tools (RBAC)	Впровадження системи контролю доступу на основі ролей
Недостатнє виявлення загроз і аномалій	Відсутність системи моніторингу загроз	Високий, після 2-3 перших місяців	Anomaly and intrusion detection tools (Snort, GuardDuty)	Впровадження IDS/IPS систем для моніторингу мережевого трафіку
Незаконний доступ до системи	Недостатній контроль доступу	Високий, протягом перших місяців	Access restriction tools (RBAC)	Впровадження системи контролю доступу на основі ролей
Відсутність системного підходу до оцінки загроз	Недостатня оцінка ризиків і загроз	Середній, після 1-3 перших місяців	Threat modeling tools (Microsoft Threat Modeling Tool)	Регулярне проведення моделювання загроз для оцінки нових функцій і змін
Недостатній аналіз і моніторинг тестування безпеки	Відсутність централізованого сховища результатів тестів	Середній, після 2-3 перших місяців	Security test results storage and monitoring tools (SaaS SIEM tools)	Використання SIEM для збору, аналізу тестування безпеки

Кінець таблиці 4.3

Загроза	Вразливість	Рейтинг та строки втілення	Інструменти та методи для усунення	Інтеграція
Вразливості в образах контейнерів	Недостатня перевірка образів контейнерів	Середній, після 2-3 перших місяців	Container image scanning tools (Snyk Container)	Регулярне сканування образів контейнерів перед розгортанням
Вразливості в операційних системах	Недостатня перевірка операційних систем	Середній, після 2-3 перших місяців	Operating system scanning tools (OpenVAS)	Регулярне сканування операційних систем для виявлення вразливостей
Вразливості в інфраструктурі	Недостатній захист інфраструктурних компонентів	Середній, після 2-3 перших місяців	Infrastructure scanning tools (ScoutSuite)	Регулярне сканування інфраструктури для виявлення вразливостей
Вразливості в коді інфраструктур	Недостатня якість коду інфраструктури з точки зору безпеки	Середній, після 2-3 перших місяців	Infrastructure code scanning tools (Chekhov)	Регулярне сканування коду інфраструктури для виявлення вразливостей

Представлена модель загроз демонструє системний підхід до ідентифікації та оцінки загроз для IT-стартапів. Використовуючи результати дослідження вразливостей і відповідні інструменти та методи DevSecOps, ця модель забезпечує

інтеграцію заходів безпеки в повсякденні процеси, мінімізуючи навантаження на команду розробників.

Впровадження такої моделі загроз дозволить стартапам забезпечити достатній рівень безпеки, використовуючи сучасні методології та інструменти для попередження і усунення загроз.

Інтеграція заходів безпеки за допомогою методів DevSecOps на ранніх етапах розробки сприяє зменшенню витрат на усунення вразливостей у майбутньому.

Використання моделі, інструментів та методів DevSecOps також позитивно впливає на обізнаність команд про потенційні ризики та методи їх усунення, що є важливим аспектом в умовах швидкого розвитку технологій і постійно змінюваного ландшафту загроз.

Крім того, запровадження моделі дозволяє побудувати комплексний захист, що охоплює різні аспекти безпеки: від аналізу компонентів системи до постійного моніторингу загроз і впровадження контрзаходів.

Таким чином, дана модель створює потенціал для забезпечення захисту від різноманітних атак та ризиків пов'язаних з майбутнім продуктом.

ВИСНОВКИ

Проведене дослідження підтвердило важливість інтеграції методів DevSecOps у процес розробки програмного забезпечення для підтримки інформаційної безпеки в IT-стартапах. Було розглянуто як теоретичні, так і практичні аспекти впровадження DevSecOps, включаючи аналіз ключових викликів, загроз та вразливостей, а також методів та інструментів для їх подолання.

Аналіз викликів інформаційної безпеки дозволив виявити, що IT-стартапи зіштовхуються з різноманітними загрозами інформаційної безпеки, які можуть мати критичні наслідки для їхньої діяльності. Головні виклики включають обмежені ресурси, швидкі зміни в технологіях та необхідність швидкої адаптації до нових загроз.

Аналіз сучасних джерел загроз та вразливостей показав основні типи загроз, що впливають на інформаційну безпеку IT-стартапів, зокрема атакуючі техніки та сценарії, які найчастіше використовуються. На основі даних CVE проаналізовано тренди за даними про сучасні та поширені вразливості.

Теоретичне дослідження методів DevSecOps виявило кілька ключових методів DevSecOps, що використовуються в сучасній розробці програмного забезпечення, що включають – зміщення безпеки вліво, безперервне тестування безпеки, інтеграцію автоматизованих інструментів та моніторинг у режимі реального часу. Розглянуто кілька практичних прикладів успішного впровадження DevSecOps у відомих компаніях, що підтвердило ефективність цих підходів.

Створено модель машинного навчання для класифікації загроз на основі даних CVE та DevSecOps інструментів. Ця модель дозволяє оцінювати актуальність методів DevSecOps та зрозуміти їх зв'язок із різними сценаріями ризику та вразливостями.

Значущість результатів:

– інформаційна безпека: впровадження методів DevSecOps дозволяє значно підвищити рівень інформаційної безпеки в IT-стартапах, знижуючи ризики пов'язані з кіберзагрозами;

– економічна ефективність: використання автоматизованих інструментів DevSecOps дозволяє зменшити витрати на виявлення та усунення вразливостей, підвищуючи ефективність процесів розробки;

– конкурентоспроможність: Забезпечення високого рівня інформаційної безпеки сприяє підвищенню довіри з боку клієнтів та інвесторів, що є важливим фактором для успішного розвитку стартапів.

Результати дослідження трендів за вразливостями та інструментами DevSecOps, розроблена модель загроз та рекомендації, щодо використання DevSecOps методів можуть бути використані IT-стартапами для покращення інформаційної безпеки та ефективності процесів розробки.

Подальше дослідження може продовжити вивчення методів DevSecOps, можливість інтеграції нових інструментів та технологій для підвищення рівня інформаційної безпеки, а також оцінити ефективність впровадження методів DevSecOps у різних типах IT-стартапів для розробки більш адаптованих рекомендацій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Routenberg J. 6 common challenges facing cybersecurity teams and how to overcome them | TechCrunch. URL: https://techcrunch.com/2023/04/06/6-common-challenges-facing-cybersecurity-teams-and-how-to-overcome-them/ (дата звернення: 10.03.2024).

2. Elsawah A. 5 Problems With Startup Security | Last Week As A vCISO. URL: https://www.lastweekasavciso.com/p/5-problems-with-startup-security (дата звернення: 11.03.2024).

3. Russel D. What Are The Security Risks of Growing Too Fast as a Startup? | Metomic. Metomic | Data Security Software for SaaS, GenAI and Cloud. URL: https://metomic.io/resource-centre/what-are-the-security-risks-of-growing-too-fast-as-a-startup (дата звернення: 12.03.2024).

4. Secure Innovation: Investor Guidance | NPSA. National Protective Security Authority | NPSA. URL: https://www.npsa.gov.uk/secure-innovation/investor-guidance (date of access: 12.03.2024).

5. Security compliance for startups: 3 reasons you need to start now. Strike Graph: Cybersecurity Compliance SaaS. URL: https://www.strikegraph.com/blog/compliance-for-startups (дата звернення: 12.03.2024).

6. What Is DevSecOps? Definition and Best Practices | Microsoft Security. Your request has been blocked. This could be due to several reasons. URL: https://www.microsoft.com/en-us/security/business/security-101/what-is-devsecops (дата звернення: 15.03.2024).

7. Kindra C. What Is DevSecOps? Exploring the Benefit & Role of DevSecOps. Springboard Blog. URL: https://www.springboard.com/blog/software-engineering/what-is-devsecops (дата звернення: 15.03.2024).

8. DevSecOps: Quick Guide to Process, Tools, and Best Practices | HackerOne. HackerOne | #1 Trusted Security Platform and Hacker Program. URL: https://www.hackerone.com/knowledge-center/devsecops-quick-guide-process-tools-an

d-best-practices (дата звернення: 13.03.2024).

9. Adabala R. Implementing DevSecOps Using AWS CodePipeline | Amazon Web Services. Amazon Web Services. URL: https://aws.amazon.com/blogs/devops/implementing-devsecops-using-aws-codepipeline/ (дата звернення: 13.03.2024).

10. Manepalli S. Building end-to-end AWS DevSecOps CI/CD pipeline with open source SCA, SAST and DAST tools | Amazon Web Services. Amazon Web Services. URL: https://aws.amazon.com/blogs/devops/building-end-to-end-aws-devsecops-ci-cd-pipeline-with-open-source-sca-sast-and-dast-tools/ (дата звернення: 15.03.2024).

11. DevSecOps Examples | Successes and Lessons Learned | Snyk. Snyk. URL: https://snyk.io/series/devsecops/share-the-journey/ (дата звернення: 16.03.2024).

12. DeVito A. 25 Top DevSecOps Tools (Ultimate Guide for 2024). StationX. URL: https://www.stationx.net/top-devsecops-tools/ (дата звернення: 16.03.2024).

13. CVE Website. CVE Website. URL: https://www.cve.org/Downloads (дата звернення: 17.04.2024).

14. Gradient boosting machines, a tutorial. PubMed Central (PMC). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3885826/ (дата звернення: 17.03.2024).

15. Smelyakov K., Filipov O. Comparative Analysis of the Applicability of Five Clustering Algorithms for Market Segmentation. 2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), м. Vilnius, Lithuania, 27 квіт. 2023 р. 2023. URL: https://doi.org/10.1109/estream59056.2023.10134796 (дата звернення: 19.04.2024).

16. What is Data Labeling? - Data Labeling Explained - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/what-is/data-labeling (дата звернення: 18.03.2024).

17. Pickard S. 13 Best DevSecOps Tools for 2024 (Paid & Free). Comparitech. URL: https://www.comparitech.com/net-admin/best-devsecops-tools/ (дата звернення: 15.04.2024).

18. Akula B. S. Vulnerability Management in DevSecOps - DZone. dzone.com. URL: https://dzone.com/articles/vulnerability-management-in-devsecops (дата звернення: 19.04.2024).

19. Cyber Threat Intelligence - VulDB. URL: https://vuldb.com/ru/?kb.cti (дата звернення: 18.03.2024).

20. Threat Modeling Process | OWASP Foundation. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/www-community/Threat_Modeling_Process (дата звернення: 18.04.2024).

21. Threats - Microsoft Threat Modeling Tool - Azure. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats#stride-model (дата звернення: 18.04.2024).